On a Method of Numerical Differentiation

Mira Bozzini and Milvia Rossini

Abstract. In this paper we present a method for the numerical differentiation of two-dimensional functions when scattered data are given. The method is based on a regularization of the given sample.

§1. Introduction

In this paper we present a method for the numerical differentiation of two-dimensional functions when scattered data are given. The problem of numerical differentiation is very important when dealing with function approximation. In fact, a satisfactory recovery of a function given in a sampled form needs some knowledge of the derivatives.

It is also well known that this problem is ill-conditioned. Consider for instance, one-dimensional equispaced data with $h_1 = 10^l$ and $h_2 = 2^p$. On a computer, because of the base change, numerical differentiation gives considerable errors in the first case, and a more accurate solution in the second case. Moreover, it is strongly influenced by the data position.

The literature on scattered data, includes the papers [7,8,10], their improvements [3,5], and some experiments on their use [9]. These papers provide the gradient approximation at the sampled points (see [7,10]) or the approximation of the gradient function (see [8]). This is done by triangulation or local and global moving least square interpolation. In some of them, asymptotic bounds for the error are also supplied.

Generally, these methods provide a satisfactory approximation inside the domain in which the data are given, but they may give large errors at the boundary (see Figs. 1–4 below).

In this paper we present a method based on a regularization of the sample which gives an error with an uniform behaviour on the whole domain in which the data are assigned. The regularization is done by constructing a new set on a regular grid. Namely, taking into account the previous observations, we have considered dyadic grids. Obviously the construction of this new set will

Curve and Surface Fitting: Saint-Malo 1999
Albert Cohen, Christophe Rabut, and Larry L. Schumaker (eds.), pp. 85–94.

ISBN 0-8265-1357-3.

generate errors that can be thought of as causal errors depending on the sample. Then (see [4]), to smooth the data, we perform a wavelet decomposition of the signal. Using the smoothed data, we approximate the gradient at the grid points by the classical finite centered difference formulas, and finally we construct a smooth function approximating the unknown gradient.

The paper is organised as follows. In §2 the method is described and the gradient estimator is constructed. In §3 the convergence properties are considered. Finally in §4 we discuss some questions related to the numerical aspects of the problem and we provide some numerical examples.

§2. The Method

Suppose we are given a set of scattered points in a domain $Q \subset \mathbb{R}^2$

$$S = \{P_i^*(x_i, y_i) | \, P_i^* \in Q, \, i = 1, \ldots, N\},$$

and the set of functional data

$$F = \{(P_i^*, f(P_i^*)), \, i = 1, \ldots, N\},$$

where $f(x, y)$ is an unknown function defined on Q. Without loss of generality, we suppose $Q = [0, 1] \times [0, 1]$.

The first step of our method consists in generating, from F, a new set of functional values, say $\hat{F}$, located on a dyadic lattice T of Q,

$$T = \{\bar{P}_{\mathbf{k}}, \, \mathbf{k} = (k_1, k_2) \in \mathbb{Z}^2, \, \mathbf{k} = 1, \ldots, 2^{\bar{n}}\},$$

that is

$$\hat{F} = \{(\bar{P}_{\boldsymbol{k}}, \hat{f}(\bar{P}_{\boldsymbol{k}})), \quad \bar{P}_{\mathbf{k}} \in T\}.$$

It is clear that the new values are affected by errors

$$\hat{f}(\bar{P}_{\boldsymbol{k}}) = f(\bar{P}_{\boldsymbol{k}}) + e(\bar{P}_{\boldsymbol{k}}).$$

Therefore we need to construct $\hat{F}$ by an efficient computational method such that the error is less or of the same order as that generated by the derivative approximation we will use. A possible strategy is to interpolate the data by a local method of a suitable order m, $m \leq \alpha$ (for instance a moving least squares technique, [6, 8]) which gives a smooth interpolating function $\hat{f}(x, y) \in C^m(Q)$ with an error $e(x, y) = O(h^m)$, where h is a local parameter depending on the distribution of the points P_i^* in Q (usually $h = 1/\sqrt{N}$).

The new set $\hat{F}$ can be thought of as a sample coming from a stochastic process depending on the points $P_i^* \in S$. Then we can use the method described in [4] for noisy data. In the next section, we will perform a wavelet decomposition in order to smooth the errors e, and we will define an estimator $\hat{g}_{j(\bar{n})}(x, y)$ of the underlying function $f(x, y)$. Then we will approximate the unknown gradient using the function estimator $\hat{g}_{j(\bar{n})}(x, y)$, and the classical approach of central finite differences.

In this paper we assume that

i) $f(x,y)$ belongs to an Hölder space of order $\alpha > 2$ on $Q^* \supset Q$, say $C^\alpha(Q^*)$.

ii) We consider a multiresolution analysis of $L^2(\mathbb{R}^2)$ given by the tensor product of two one-dimensional s-regular multiresolution analysis such that the scaling function ϕ is a coiflet, that is a compactly supported orthonormal function such that the scaling function $\phi(x)$ and the wavelet $\psi(x)$ have $L-1$ and L, vanishing moments respectively. Moreover, we assume that $L > [\alpha]+1$ and $s > \alpha$.

2.1. The function estimator

As mentioned above, for approximating the gradient of $f(x,y)$, we will use the function $\hat{g}_{j(\bar{n})}(x,y)$ defined as in [4]. In this section we briefly describe its definition and the motivations that lead to consider this function.

It is known that when we perform a MRA using data given on a subset of $\mathbb{R}^2$, we need to take into account the problem of reducing the boundary errors. To this end, we consider a function $g(x,y) \in C^\alpha(\mathbb{R}^2)$ compactly supported on $Q^* \supset Q$ such that

$$g(x,y) = f(x,y), \quad \forall (x,y) \in Q,$$

and a new dyadic lattice T^* on Q^* of dimension $2^n \times 2^n$, such that $T^* = \{T \cup \{ \text{ points sampled in } Q^* \setminus Q\}\}$. Moreover, the advantage of the nested structure of a MRA is that to provide an efficient tree-structure algorithm for the decomposition of functions in $\boldsymbol{V}_n$ for which the smoothing coefficients $\langle g, \Phi_{n,\boldsymbol{k}} \rangle$ are given.

In applications, a function is given in sampled form, and it is therefore necessary to approximate the projection $P_{\boldsymbol{V}_n}$ on the space $\boldsymbol{V}_n$, by some operator Π_n, and to derive a reasonable estimator of Π_n in terms of the sampled values. The choice of Π_n and of its estimator is suggested by the following facts (see [1,4]).

The set of nonzero coefficients $\langle g, \Phi_{n,\boldsymbol{k}} \rangle$ has cardinality equivalent to $O(2^{2n})$. Moreover,

$$\left| \langle g, \Phi_{n,\boldsymbol{k}} \rangle - 2^{-n} g(P_{\boldsymbol{k}}) \right| \leq C 2^{-n} 2^{-n\alpha}, \tag{1}$$

where C is a constant depending on the smoothing function $\Phi(x,y)$ and on $g(x,y)$.

As a consequence, we define

$$(\Pi_n g)(x,y) = 2^{-n} \sum_{P_{\boldsymbol{k}} \in T^*} g(P_{\mathrm{k}}) \Phi_{n,\mathrm{k}}(x,y). \tag{2}$$

Since we have data corrupted by the interpolation errors, we consider the estimator of Π_n

$$(\hat{\Pi}_n g)(x,y) = 2^{-n} \{ \sum_{\bar{P}_{\boldsymbol{k}} \in T} \hat{f}(\bar{P}_{\boldsymbol{k}}) \Phi_{n,\boldsymbol{k}}(x,y) + \sum_{P_{\mathrm{k}} \in T^* \setminus T} g(P_{\mathrm{k}}) \Phi_{n,\mathrm{k}}(x,y) \}. \tag{3}$$

This estimator may lead to an oscillatory solution bearing too much fidelity to the data. Since, for numerical differentiation, we need to correctly smooth the data, we have to associate to each sample of size $2^{\bar{n}} \times 2^{\bar{n}}$ a resolution $j(\bar{n}) < \bar{n}$, and to consider the orthogonal projection of $(\hat{\Pi}_n g)(x,y)$ onto $\boldsymbol{V}_{j(\bar{n})}$, that is

$$\hat{g}_{j(\bar{n})}(x,y) = (P_{\boldsymbol{V}_{j(\bar{n})}} \hat{\Pi}_n g)(x,y). \tag{4}$$

The parameter $j(\bar{n})$ governs the smoothness of our estimator, and it is important to choose it in the right way because it controls the tradeoff between the fidelity to the data and the smoothness of the resulting solution. From a theoretical point of view, the smoothing parameter must tend to infinity at the correct rate, as the amount of information in the data grows to infinity.

2.2. The gradient estimator

We now consider the construction of the gradient estimator. When dealing with gridded data, it is natural to approximate the gradient using the usual centered difference formulas. Let

$$(grad\, f)(x,y) = \left(f^{(x)}(x,y), f^{(y)}(x,y)\right),$$

be the gradient of $f(x,y)$, and let D_r^x, D_r^y be the centered difference operators which use r equispaced points in the x or y direction respectively. If $r \leq [\alpha]$, we know that

$$(D_r^t f)(\bar{P}_{\boldsymbol{k}}) = f^{(t)}(\bar{P}_{\boldsymbol{k}}) + O(2^{-n(r-1)}), \quad t = \{x,y\}.$$

Then, at each point of the lattice T, we approximate $(grad\, f)(\bar{P}_{\boldsymbol{k}})$ by

$$\widehat{(grad\, f)}(\bar{P}_{\mathbf{k}}) = \left((D_r^x \hat{g}_{j(\bar{n})})(\bar{P}_{\mathbf{k}}), (D_r^y \hat{g}_{j(\bar{n})})(\bar{P}_{\mathbf{k}})\right). \tag{5}$$

Using the data (5), it is possible to define, at each point of Q, an estimator of $f^{(t)}(x,y)$, $t = \{x,y\}$, by the operator $\hat{\Pi}_n$. Namely, we consider the function $g(x,y)$, and we define

$$\widehat{(grad\, f)}(x,y) = \left((\hat{\Pi}_n g^{(x)})(x,y), (\hat{\Pi}_n g^{(y)})(x,y)\right), \tag{6}$$

where

$$(\hat{\Pi}_n g^{(t)})(x,y) = 2^{-n}\{ \sum_{\bar{P}_{\mathbf{k}} \in T} (D_r^t \hat{g}_{j(\bar{n})})(\bar{P}_{\mathbf{k}}) \Phi_{n,\mathbf{k}}(x,y) + \sum_{P_{\mathbf{k}} \in T^* \setminus T} g^{(t)}(P_{\mathbf{k}}) \Phi_{n,\mathbf{k}}(x,y)\}.$$

§3. Asymptotic Properties

Under the assumptions of Section 2, and taking into account the properties of the projection $P_{\mathbf{V}_l}$, we know that

$$|(P_{\mathbf{V}_l}g)(x,y) - g(x,y)| = O(2^{-l\alpha}), \quad \forall (x,y) \in Q^*. \tag{7}$$

Moreover, from (1) we have

$$|(\Pi_l g)(x,y) - (P_{\mathbf{V}_l}g)(x,y)| = O(2^{-l\alpha}) \quad \forall (x,y) \in Q^*, \tag{8}$$

then

$$|(\Pi_l g)(x,y) - f(x,y)| = O(2^{-l\alpha}) \quad \forall (x,y) \in Q. \tag{9}$$

Using relations (7), (8), (9) and the results stated in [4], we have proved

Proposition 1. *If assumptions i) and ii) of Section 2 hold, we have*

$$|\hat{g}_{j(\bar{n})}(x,y) - f(x,y)| = O(2^{-j(\bar{n})\alpha}) + O(h^m),$$

for every $(x,y) \in Q$.

Remark 1. *This result points out how the choice of* $j(\bar{n})$ *depends on the sample dimension* N *and on* m. *In fact it has to be chosen so that* $2^{-j(\bar{n})\alpha}$ *is less or of the same order of* h^m.

Proposition 2. *If assumptions i), ii) of Section 2 hold, the approximation (6) of the gradient satisfies, asymptotically, the following bound*

$$\begin{aligned} |(\hat{\Pi}_n g^{(t)})(x,y) - f^{(t)}(x,y)| =& O(2^{-j(\bar{n})\alpha}) + O(2^{-n(r-1)}) \\ &+ O(h^m) + O(2^{-n(\alpha-1)}), \end{aligned}$$

for every $(x,y) \in Q$.

§4. Numerical Results

In this section, we discuss some questions related to the computational costs and to the numerical implementation of the method we have studied. We also present some numerical results.

4.1. Computational costs

The computational costs are essentially given by the wavelet decomposition and by the construction of the gridded data set $\hat{F}$ of dimension 2^{2n}. For the wavelet decomposition, they are at the most of the same order of the sample dimension, that is $O(2^{2n})$. For the construction of $\hat{F}$, if we use a local method of order $m \ll N$, they are given by the solution of N linear systems of dimension $2m+1$. Then the computational costs will be $O\big((\frac{2m+1}{3})^3 N\big) + O(2^{2n})$.

4.2. Numerical implementation

In this section we discuss some questions related to the construction of $\hat{g}_{j(\bar{n})}(x,y)$ and $(\hat{\Pi}_n g^{(t)})(x,y)$. Following the idea of §3.1, we need to extend the given signal to a suitable square $Q^* \supset Q$

$$Q^* = [-2^{-\bar{n}}K, 1+2^{-\bar{n}}K] \times [-2^{-\bar{n}}K, 1+2^{-\bar{n}}K],$$

forcing it to be zero at the Q^* boundary. This is necessary in order to avoid undesirable behaviour at the boundary. The extension can be done following a method proposed in [2]. Note that K is related to the number of points we consider outside Q. On one hand, it cannot be chosen too small, otherwise undesirable boundary oscillations could occur. On the other hand, it depends on the resolution level $j(\bar{n})$. In fact, it has to be chosen so that the discrete wavelet decomposition can be performed.

In the numerical examples we have used the coiflets with $L = 6$ vanishing moments. For the pointwise gradient approximation, we have used the central finite difference of order 4 ($r = 5$).

4.3. Numerical examples

In this paper we present the results achieved for the test functions

$$\begin{aligned} f_1(x,y) &= 0.75\exp[-((9x-2)^2+(9y-2)^2)/4] \\ &\quad + 0.75\exp[-((9x+1)^2/49+(9y+1)^2/10)] \\ &\quad + 0.5\exp[-((9x-7)^2+(9y-3)^2)/4] \\ &\quad - 0.2\exp[-((9x-4)^2+(9y-7)^2)], \\ f_2(x,y) &= \frac{1}{\sqrt{(1+2\exp(-2\sqrt{100x^2+100y^2}-6.7)}}, \end{aligned}$$

defined on the unit square $[0,1]\times[0,1]$. We have considered N scattered points on Q, and have constructed a new gridded data set $\hat{F}$ of size $2^{\bar{n}} \times 2^{\bar{n}}$ using the modified quadratic Shepard method. The smoothing parameter $j(\bar{n})$ has been chosen taking into account Remark 1 of Section 3. Therefore, having used the modified quadratic Shepard method with $\bar{n} = 5$, a possible choice is $j(\bar{n}) = 4$.

We now present our results, and compare them with those obtained with the method (L-method) proposed in [8] which, among those we find in the literature, we belive is preferable both for its theoretical aspects and numerical performance.

The following examples show how the proposed method provides an approximation which seems to have the same behaviour for the functions considered, both for graphical results and for errors. For the sake of brevity, we show only the approximations of one gradient component, but give the relative errors for both of them.

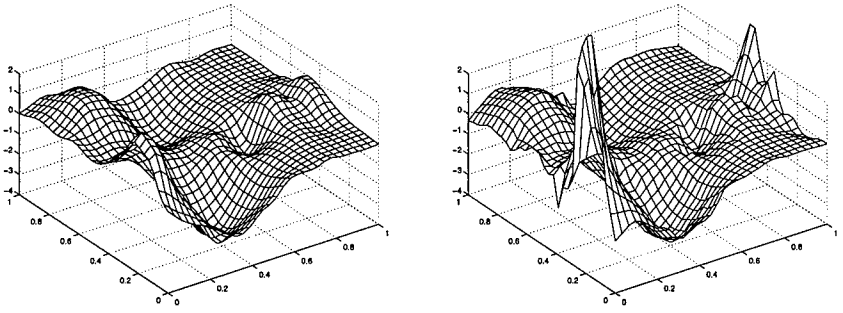

Fig. 1. Example 1, $N = 100$. On the left-hand side, the result of our method. On the right-hand side, the result of the L-method.

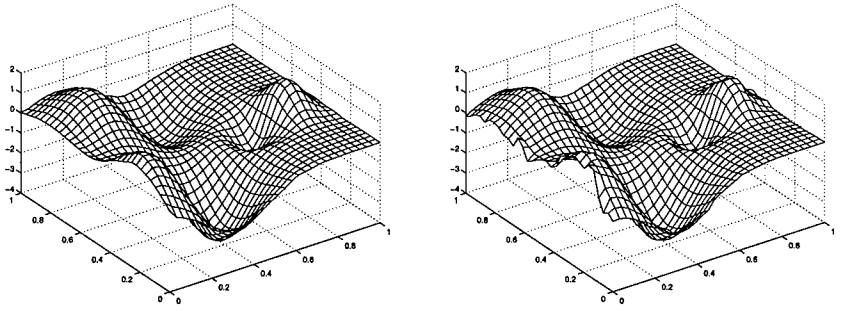

Fig. 2. Example 1. $N = 300$. On the left-hand side, the results of our method. On the right-hand side, the result of the L-method.

Example 1. Consider $N = 100$ and $N = 300$ values of $f_1(x, y)$. In Figs. 1 and 2 we present the y-partial derivative approximations. The following table lists er:= relative error of our method and erL:= relative error of the L-method:

f_1	$N = 100$		$N = 300$	
	er	erL	er	erL
f^x	24%	55%	6%	12%
f^y	43%	226%	36%	36%

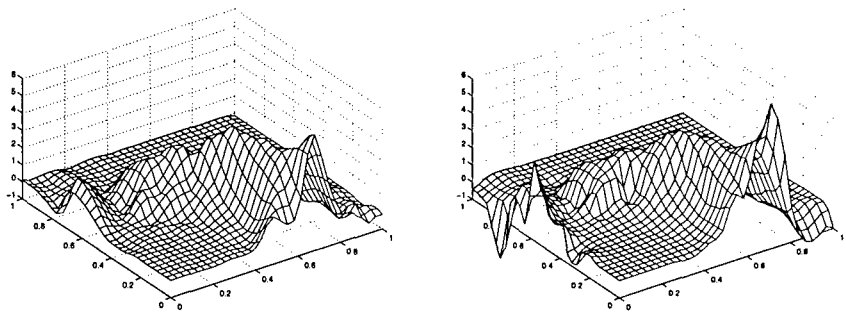

Fig. 3. Example 2. $N = 100$. On the left-hand side, the result of our method. On the right-hand side, the result of the L-method.

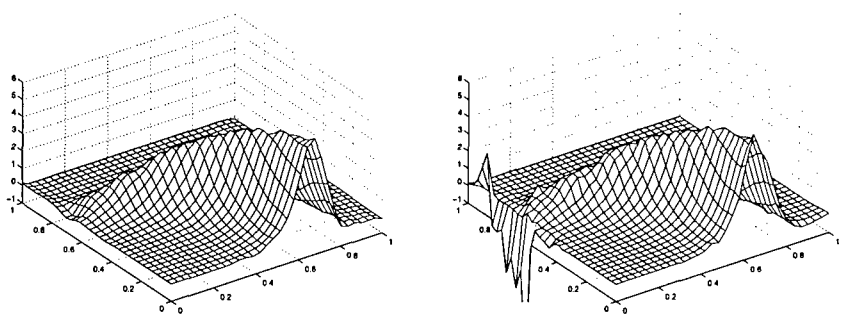

Fig. 4. Example 2. $N = 300$. On the left-hand side, the result of our method. On the right-hand side, the result of the L-method.

Example 2. Consider $N = 100$ and $N = 300$ values of $f_2(x, y)$. In Figs. 3 and 4 we present the results achieved for the y-partial derivative, and in Fig. 5 the error functions for $N = 300$. The following table lists er:= relative error of our method, and erL:= relative error of the L-method:

f_2	$N = 100$		$N = 300$	
	er	erL	er	erL
f^x	35%	400%	6%	54%
f^y	27%	64%	6%	135%

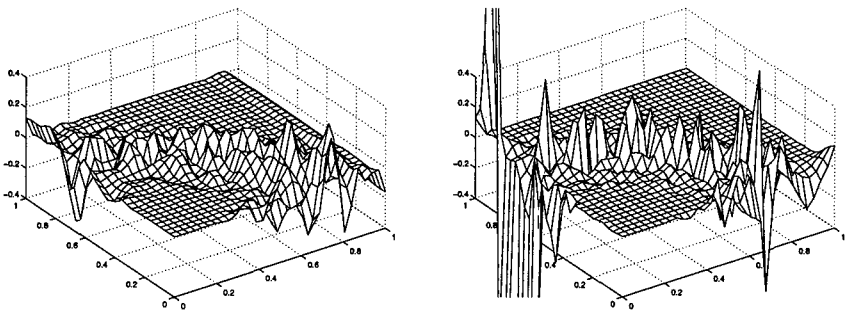

Fig. 5. Example 2. $N = 300$. On the left-hand side, the error function of our method. On the right-hand side, the error function of L-method.

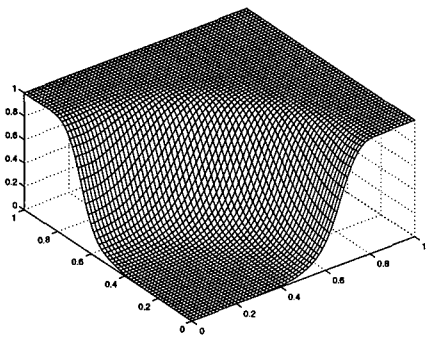

Fig. 6. Example 2. $N = 300$. The approximation of $f_2(x, y)$. The absolute maximum error is 0.015.

Finally, as is usual in the literature, we consider an application to function recovery which shows the goodness of the gradient approximation. We recover $f_2(x, y)$ by Hermite interpolation at 16×16 nodes, where we interpolate the data coming from the estimator $\hat{g}_{j(\bar{n})}(x, y)$ and from the approximated gradient (Fig. 6).

References

1. Antoniadis A., Wavelet methods for smoothing noisy data, in *Curves and Surfaces in Geometric Design*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), A. K. Peters, Wellesley MA, 1994, 485–492.
2. Bozzini M. and M. Rossini, Approximating surfaces with discontinuity, Mathematical and Computer Modelling, to appear.

3. De Marchi S., On Computing derivatives for C^1 interpolating schemes: an optimization. Computing **60** (1998), 29–53.
4. Bozzini M. and M. Rossini, Numerical differentiation of 2D functions from noisy data, preprint 1999.
5. Feraudi F. and F. De Tisi, A local Method for shape preserving partial derivatives, Technical Report of the Department of Mathematics, Milano University, 1999.
6. Franke R. and G. M. Nielson, Smooth interpolation of large scattered data, Int. J. for Numerical Methods in Engineering **38** (1980), 1691–1704.
7. Goodmann T. N. T., H. B. Said, and L. H. T. Chang, Local derivative estimation for scattered data, App. Math. Comp. **68** (1995), 41–50.
8. Levin D., The approximation power of moving least-squares, Math. Comp. **67**, n. 224 (1998), 1517–1531.
9. Moriconi S., Sull'approssimazione numerica delle derivate e loro utilizzo nella ricostruzione di superfici da dati sparsi. Math. Degree Thesis, University of Milano, 1997.
10. Renka R. J. and A. K. Cline, A triangle based C^1 interpolation method, Rocky Mountain J. of Mathematics **14** (1984), 223–237.

Mira Bozzini
Dipartimento di Matematica e Applicazioni
Università di Milano Bicocca
via Bicocca degli Arcimboldi 8
20126, Milano Italy
bozzini@matapp.unimib.it

Milvia Rossini
Dipartimento di Matematica e Applicazioni
Università di Milano Bicocca
via Bicocca degli Arcimboldi 8
20126, Milano Italy
rossini@matapp.unimib.it